

Національний лісотехнічний університет України

(повне найменування вищого навчального закладу)

Навчально-науковий інститут комп'ютерних наук

та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

Магістерська кваліфікаційна робота

другий (магістерський)

(рівень вищої освіти)

на тему: "Розроблення мультиплеєрного застосунку

для вивчення іноземних мов засобами Unity"

Виконав: студент 6 курсу групи КН-62М
спеціальності

122 "Комп'ютерні науки"

(шифр і назва напрямку підготовки, спеціальності)

Шелепетень І.В.

(прізвище та ініціали)

Керівник Борецька І.Б.

(прізвище та ініціали)

Рецензент Яцишин С. І.

(прізвище та ініціали)

Львів – 2024

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

ННІ комп'ютерних наук та інформаційних технологій

Кафедра комп'ютерних наук

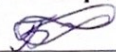
Рівень вищої освіти другий (магістерський)

Спеціальність 122 "Комп'ютерні науки"

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри


Борецька І.Б.
"05" січня 2024 року

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Шелепетень Іван Васильович

(прізвище, ім'я, по батькові)

1. Тема роботи: "Розроблення мультиплеєрного застосунку для вивчення іноземних мов засобами Unity"

керівник роботи Борецька Ірина Богданівна, к.т.н, доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від "13" лютого 2023 року

№ C-49

2. Термін подання студентом роботи "05" січня 2024 року

3. Вихідні дані до роботи: документація Unity, документація Asp.Net, документація Entity Framework

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

1. Стан проблемної області

2. Інформаційне забезпечення

3. Математичне забезпечення

4. Програмне забезпечення

5. Розроблення стартап-проекту

6. Висновки

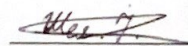
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання "15" лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Отримання завдання	15.02.2023	Виконано
2.	Збір інформації про побудову API	25.02.2023	Виконано
3.	Створення API	23.04.2023	Виконано
4.	Підключення бази даних	01.05.2023	Виконано
5.	Ознайомлення зі створенням ТСП-з'єднання	12.05.2023	Виконано
6.	Створення клієнт-серверного з'єднання	07.08.2023	Виконано
7.	Побудова UI	15.10.2023	Виконано
8.	Тестування продукту	11.12.2023	Виконано
9.	Розроблення стартап-проекту	22.12.2023	Виконано
10.	Підготовка пояснюючої записки	30.12.2023	Виконано
11.	Здача проекту	05.01.2024	Виконано

Студент


(підпис)

Шелепетень І.В.
(прізвище та ініціали)

Керівник роботи


(підпис)

Борецька І.Б.
(прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота містить 40 сторінки пояснювальної записки, 32 рисунки, 1 формули та 10 джерел інформації.

Підчас виконання даної дипломної роботи, було створено мобільний застосунок для вивчення іноземних мов на базі ігрового рушія Unity та сервер для обробки логіки програми на основі Asp.Net. З'єднання між застосунком та сервером налаштовано з допомогою API та TCP-з'єднання.

Ключові слова: *Unity, Asp.Net, TCP-з'єднання, сервер, мультиплеєр, JSON, UiToolkit.*

ABSTRACT

Graduate work contains 40 pages of explanatory note, 32 figures, 1 mathematical formula and 10 used literary sources.

During the implementation of this graduate work, a mobile application for learning foreign languages based on the Unity game engine and a server for processing program logic based on Asp.Net were created. The connection between the application and the server is configured using an API and a TCP connection.

Keywords: *Unity, Asp.Net, TCP connection, server, multiplayer, JSON, UiToolkit.*

ТЕХНІЧНЕ ЗАВДАННЯ

Необхідно розробити мультиплеєрну платформу для вивчення іноземних мов, а саме:

- Розробити сервер з доступом до баз даних і реалізувати роботу з ними через WebAPI
- На сервері спроектувати мультиплеєр для можливості з'єднання користувача з іншими за пинципом Клієнт-Сервер.
- Розробити мобільний застосунок, який буде отримувати інформацію з баз даних за допомогою HTTP-запитів
- Створити користувацький інтерфейс, методами Unity Ui Toolkit
- Винести весь ігровий процес за межі мобільного застосунку та реалізувати його в мультиплеєрі
- Встановити стабільне TCP-з'єднання з мультиплеєром

ЗМІСТ

АНОТАЦІЯ.....	3
ТЕХНІЧНЕ ЗАВДАННЯ.....	4
ЗМІСТ.....	5
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1: СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	8
1.1 Огляд проблемної області.....	8
1.2 Перспективи розвитку.....	10
РОЗДІЛ 2: ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	11
2.1 Unity.....	11
2.2 Порівняння Unity та Unreal Engine.....	12
2.3 Порівняння Unity та Godot.....	16
РОЗДІЛ 3: МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	20
3.1 Математичні операції в проекті.....	20
РОЗДІЛ 4: ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.....	21
4.1 Послідовний опис розробки проекту.....	21
4.2 Тестування програми.....	26
РОЗДІЛ 5: РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ.....	31
5.1 Опис ідеї проекту.....	31
5.2 Аналіз технологічних можливостей реалізації проекту.....	32
5.3 Аналіз ринкових можливостей запуску стартап-проекту.....	32
5.4 Розроблення ринкової стратегії проекту.....	33
5.5 Маркетингова програма стартап-проекту.....	33
5.6 Висновки до розділу.....	34
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	37
ДОДАТКИ.....	38

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

Мультиплеєр – застосунок, який дозволяє взаємодії між користувачами.

Кросплатформенний – застосунок, що може працювати на різних операційних системах.

UI – користувацький інтерфейс.

TCP-з'єднання – стабільне двостороннє з'єднання між двома комп'ютерами.

Серіалізація – приведення об'єкта класу в текстовий формат.

Десеріалізація – перетворення тексту в об'єкт певного класу.

AAA ігри – ігри з капіталовкладенням більше 20млн. доларів

ВСТУП

Пік розвитку застосунків для вивчення іноземних мов припав на 2010-2015 роки. Саме тоді утворились найбільші гіганти в цій сфері, які зайняли більшу частину ринку. Причиною такого росту стала популяризація смартфонів по всьому світу, а також гостра необхідність знань іноземних мов для охоплення більшої площі інфопростору. Самі ж компанії, які розвивали такі застосунки, взяли за основу принцип: вчись будь-де та будь-коли. Один урок на смартфоні триває близько 5 хвилин, тому це дуже зручно для людей, у яких щільний графік. Знайти час на кілька перерв тривалістю в 3-5 хвилин набагато легше, ніж виділити час на 2 години заняття з репетитором. Нажаль, також, дані компанії взяли за старі методики навчання, але прикрили їх блискучим інтерфейсом. Все ті ж шкільні тести та самостійні роботи тепер в телефонах та в красивій обгортці.

Сьогоднішній ринок застосунків для вивчення іноземних мов розвивається лише в плані додаткових послуг: тимчасових подій, випробувань, тощо.

Об'єктом дослідження: розробка мультиплеєрного застосунку для вивчення іноземних мов з допомогою ігрового рушія Unity.

Метою роботи є розробка платформи, яка дозволить ефективніше вивчати іноземні мови в компанії з незнайомими людьми чи друзями.

Предметом дослідження є реалізація мультиплеєрних ігор на мові C# з використанням ігрового рушія Unity та сервера на базі Asp.Net фреймворка.

Практичне значення роботи полягає у зацікавленні до навчання та налаштування простого і невимушеного навчального процесу.

РОЗДІЛ 1: СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1 Огляд проблемної області

Останні десять років, онлайн платформи для вивчення іноземних мов є одними з найпоширеніших застосунків на мобільних телефонах користувачів по всьому світу. Причиною цього стала глобалізація інформаційних ресурсів після винайдення всесвітньої мережі інтернету та доступність подорожей між країнами. Вони стали причиною поширення іноземних мов та їх необхідності для більшості світового населення. Якщо поглянути на тенденції в сфері освіти, то популярність іноземних мов зростає в рази.

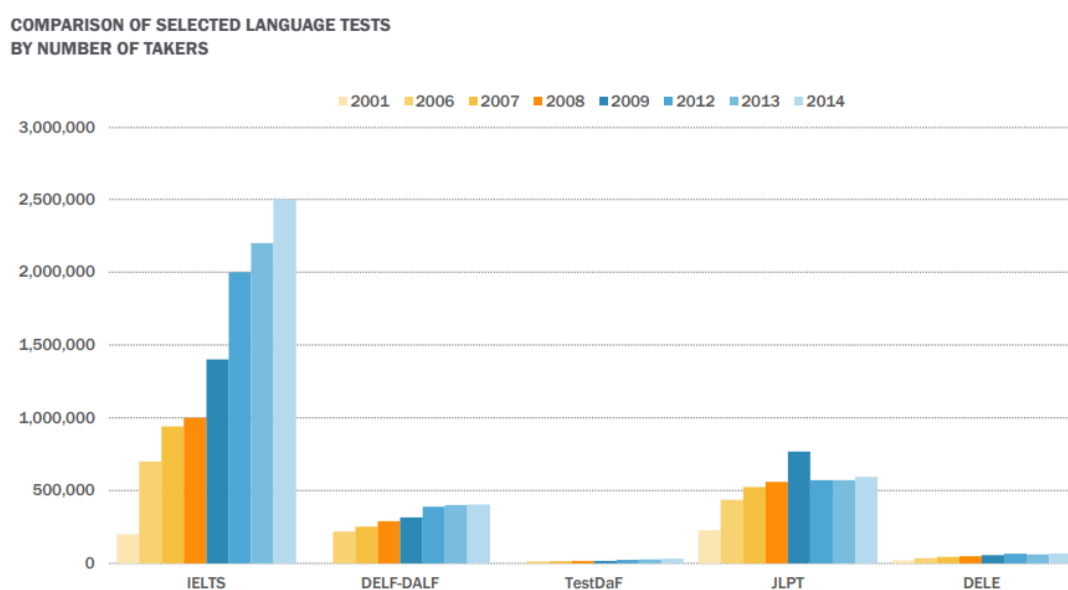


Рис.1 – Статистика здачі тестів з іноземних мов 2001-2014

Як можна побачити на діаграмі, то популярність тестів на отримання рівня володіння іноземними мовами зростає дуже різко. Особливо варто звернути увагу на тести англійської мови – IELTS. Показник популярності зріс в 10 разів. Причиною цього стало визнання англійської мови, як міжнародної, що посилює її необхідність в діловій, політичній та туристичній сферах. Ця тенденція і спричинила ріст популярності застосунків для вивчення іноземних мов.

Якщо взяти найпопулярніші платформи даної тематики, то за 10-15 років вони отримали сотні мільйонів скачувань. Наприклад, кількість завантажень Duolingo – топ-1 платформи в маркеті Google Play, складає більше 500 мільйонів. Якщо припустити, що 1/10 частина користується застосунком активно протягом року, то

виходить, більше 50 мільйонів користувачів лише одного застосунку, що тільки підкреслює популярність іноземних мов в світі.

Duolingo: уроки іноземної мови

Duolingo

Містить рекламу · Покупки через додаток

Вивчайте англійську, іспанську, французьку, німецьку та інші мови безкоштовно.



4,7★
20,8 млн відгуків

500 млн+
Завантаження

E
Для всіх вікових груп ⓘ

Рис. 2 – Статистика активності застосунку Duolingo

Варто зауважити, що застосунки, які не входять в топ-10 освітніх платформ також мають 10+ мільйонів установок.

LingoDeer - Learn Languages

LingoDeer - Learn Languages Apps

Покупки через додаток

Залишайтеся чіткими, вивчаючи японську, корейську, іспанську, французьку чи німецьку мови!



4,5★
435 тис. відгуків

10 млн+
Завантаження

E
Для всіх вікових груп ⓘ

Установити

Рис. 3 – Статистика активності застосунку LingoDeer

Причин популярності таких додатків є декілька:

- доступність – уроки в таких застосунках завжди під рукою, оскільки в сучасному світі мобільний телефон є необхідним атрибутом.
- періодичність – дані платформи дозволяють засвоїти невеликий об'єм інформації в відрізок часу близько 5хв, тому вони корисні для вбивання часу в чергах, по дорозі на роботу чи інших ситуаціях, де необхідно очікувати певний період часу. Це набагато зручніше, ніж працювати з репетитором протягом кількох годин.

- нав'язливість – застосунки для вивчення іноземних мов просто не дадуть вам забути про ваші уроки, використовуючи різні методи: від сотень повідомлень від розлюченої сови на пошту та в push-сповіщеннях (впливаючих сповіщеннях смартфона) до різних віртуальних нагород, челенджів та досягнень.
- постійність засвоєння – вивчені слова все ще будуть з'являтися в навчальній програмі для їх постійного закріплення, що дозволить не забувати пройдений матеріал.

Проте, хоча застосунки для вивчення іноземних мов мають явні переваги, їх методики подачі матеріалу залишаються достатньо одноманітними. Попри конкуренцію між компаніями, їх продукт все ще зводиться до тестів та вікторин, які аж ніяк не відносяться до їх слоганів: “вивчай, граючи”.

1.2 Перспективи розвитку

На даний момент ринок платформ для вивчення іноземних мов сталий. Застосунки освоїли механіки по-типу вікторин, голосового вводу, тощо і перестали розвиватись в даному напрямку. Зараз розвиток і успіх таких мобільних програм залежить від обгортки, в яку вони можуть загорнути дані механіки. До таких обгорток належать дизайн, події, випробування та досягнення. Тому попри велику базу навчального матеріалу, такі платформи не можуть втримати більшість користувачів, оскільки більшість відвідувачів таких застосунків – люди, які скачали їх на хвилинному бажанні опанувати нову мову. Щоб їх втримати, потрібно цікавіші механіки в основі навчальних вправ. Перспективним нововведенням може бути створення конкуренції між користувачами напряму в кожній грі, а не лише в рейтинговій таблиці в загальному. Це дозволить змагатись користувачам з суперниками рівним з ними по вміннях, що показуватиме їх покращення при постійних іграх, шляхом підвищення рейтингу. Таке оцінювання дасть краще відчуття прогресу гравцям і зацікавить більше, ніж проходження певних уроків. В такому випадку користувачі будуть серйозніше підходити до навчання і в застосунках буде ненав'язлива система оцінювання їх знань.

РОЗДІЛ 2: ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Unity

Unity — це ігровий рушій для розробки багатоплатформенних застосунків. Такі програми можна легко портувати на будь-яку операційну систему чи девайс.

Редактор Unity має інтерфейс, що складається з різних вікон, які можна розташувати на свій розсуд. Завдяки цьому можна проводити налагодження гри чи застосунка прямо в редакторі. Головні вікна — це оглядач ресурсів проєкту, інспектор поточного об'єкта, вікно попереднього перегляду, оглядач сцени та оглядач ієрархії ресурсів.

Проєкт в Unity складається з різних сцен (рівнів), які представлені окремими файлами. Кожна сцена містить власний ігровий світ з об'єктами, сценаріями і налаштуваннями. Об'єкти в сценах можуть бути моделями (ландшафт, персонажі, предмети тощо) або порожніми ігровими об'єктами, які визначають поведінку інших об'єктів (тригери подій, точки збереження прогресу тощо). Їх можна розташовувати, обертати, масштабувати і до них можна застосовувати скрипти. Кожен об'єкт має назву, тег і шар для відображення на екрані. Кожен предмет на сцені має компонент Transform, який зберігає координати, поворот і розміри. Об'єкти з видимою геометрією також можуть мати компонент Mesh Renderer, який робить модель видимою. Моделі об'єднуються в ассети для швидкого доступу.

Unity підтримує фізику твердих тіл і тканини, а також Ragdoll для реалістичного моделювання руху об'єктів. Є система успадкування об'єктів, де дочірні об'єкти наслідують зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти прикріплюються до об'єктів як компоненти.

У 2D іграх Unity використовуються спрайти, а в 3D - тривимірні моделі (меші). Текстури, матеріали і шейдери використовуються для створення зовнішнього вигляду об'єктів. Системи часток використовуються для відображення різних субстанцій, таких як рідини чи дим.



Рис. 4 – Логотип Unity

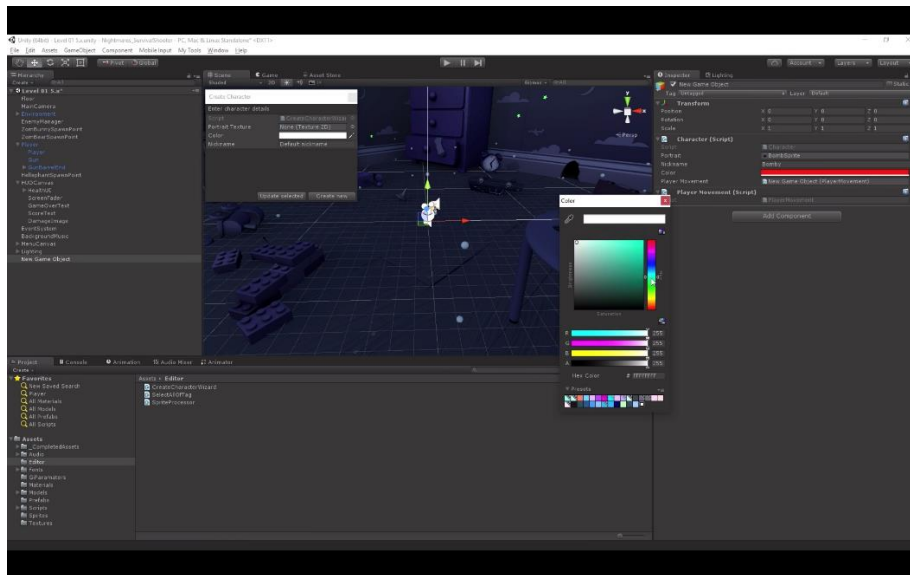


Рис. 5 – Середовище розробки Unity

2.2. Порівня Unity та Unreal Engine

На сьогоднішній день найбільш популярними рушіями для розробки ігор є Unity та Unreal Engine. Однак для визначення, який із них кращий, важливо порівняти їхні базові характеристики, напрямки та історію створення.

Щодо частоти оновлень:

- Unreal Engine вийшов у 1998 році і зазнав дев'ятох оновлень. Найсучаснішою версією є Unreal Engine 4, але також оголошено розробку п'ятої версії.
- Unity випустив одинадцять повноцінних версій з часу свого початку, останньою з яких є Unity 2024. Крім цього, регулярно виходять оновлення з невеликими покращеннями або налагодженням роботи.

У даному аспекті Unreal Engine має перевагу, оскільки розробники не повинні адаптуватися до частих нововведень, і цей рушій присутній на ринку тривалий час. Однак Unity виграє в ефективності усунення неполадок та швидкості розвитку.

Щодо цілей випуску:

- Unreal Engine спрямований на створення високоякісних шутерів, але відмічено його універсальність для розробки ігор різних жанрів, зокрема AAA-проектів.
- Unity був розроблений як безкоштовна альтернатива для інди-розробників. Його графічні можливості менше вибагливі, але рушій зручний для розробки 2D ігор та проектів віртуальної та доповненої реальності.

Необхідно відзначити, що Unreal Engine вражає красивою графікою в будь-якій грі, з великою увагою до візуальних аспектів. Unity ж володіє швидким розвитком, а також підлаштований для створення проектів віртуальної та доповненої реальності.



Рис. 6 – Інди-ігра Little Nightmares



Рис. 7 – S.T.A.L.K.E.R. 2: Heart of Chernobyl



Рис. 8 – ArcheAge II

Хоча графіка в іграх на Unity гірша, це мається на увазі лише початкові налаштування рендеру зображення. Хороший програміст може дотягнути гру до рівня AAA проєктів з мільйонним підходом до візуальних ефектів. Дані ігри чітко показують, що в Unity можна налаштувати ефекти освітлення, але також помітно, що систему часток (для створення пилу, туману, диму) стараються не використовувати. В старіших версіях система часток дуже сильно навантажувала відеокарту, тому її старалися використовувати якомога менше. Це змінилося після

випуску VFX Graphs – системи, яка дозволяє створювати висококласні, легкі для пристроїв шейдери та системи часток.



Рис. 9 – Ori and the Will of the Wisps

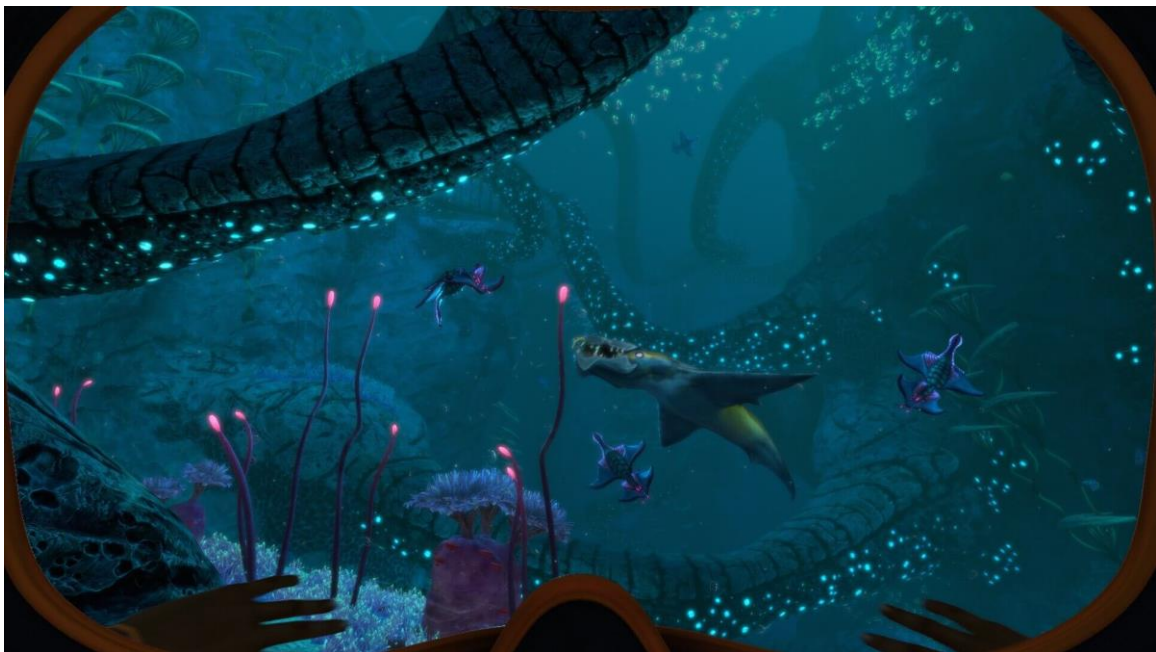


Рис. 10 – Subnautica : Below Zero

Технічні можливості:

- Unreal Engine відразу підтримує мову програмування C++. Крім того, за допомогою спеціальних адаптацій можна використовувати C#, Python і JavaScript, хоча це не завжди зручно. Для полегшення роботи новачкам, які не володіють складними мовами програмування, у русії також існує система візуального програмування Blueprints.

- Unity дозволяє розробляти застосунки на C#, JavaScript та Boo. Відзначається наявністю системи візуального програмування Bolt, хоча вона не завжди користується популярністю через недоробленість.

Переваги Unreal Engine полягають у наявності робочої системи візуального програмування та можливості підключення різних мов програмування. У свою чергу, Unity виграє у зручності використання кількох мов, які є більш доступними для новачків.

Комюніті:

- Користувачами Unreal Engine переважно є великі компанії, кожна з яких має свої модифікації для рушія, що може ускладнювати спільну розробку. Також це впливає на доступність навчального матеріалу, більшість якого платна.
- У Unity розвинуте та доступне комуніті з багатьма форумами, каналами на YouTube та різноманітними навчальними матеріалами. Документація Unity відстає за якістю від Unreal Engine, але з компенсацією надається великою кількістю безкоштовних курсів для новачків.

Узагальнюючи, можна сказати:

- Unity є простішим та більш дружнім до новачків, має швидко розвиваючися комуніті і підходить для інді-розробників, особливо в галузі мобільної та віртуальної реальності.
- Unreal Engine славиться графікою та використовується для розробки AAA-ігор, але вимагає більше зусиль для вивчення та пристосування новачків до складних мов програмування.

Усе залежить від конкретних потреб і навичок розробника при виборі між Unity та Unreal Engine.

2.3. Порівня Unity та Godot

Godot - це рушій для розробки 2D та 3D застосунків, який був створений у 2014 році. На відміну від Unity, Godot ще не зайняв конкретну нішу в галузі розробки, оскільки продовжує активно розвиватися. Однією з його переваг порівняно з конкурентами є можливість створювати всі елементи гри прямо у рушії,

без необхідності користування зовнішніми програмами. В середовищі Godot можна створити графіку та музику з нуля, що відрізняє його від Unreal Engine та Unity. Додатковим плюсом є те, що весь код основи цього рушія є відкритим, що надає розробникам можливість створювати власні версії рушія.

Основною мовою для написання скриптів в Godot є GDScript – мова програмування, схожа на Python, спеціально розроблена для цього рушія. Завдяки відкритому ядру рушія існує також версія Godot Engine Mono Version, яка підтримує C#. Крім того, через систему GDNative є можливість підключати мови програмування, такі як C++, D та Rust, і розробники вказують, що працюють над інтеграцією зовнішніх мов програмування, таких як Lua, Python і Squirrel.



Рис. 11 Логотип Godot

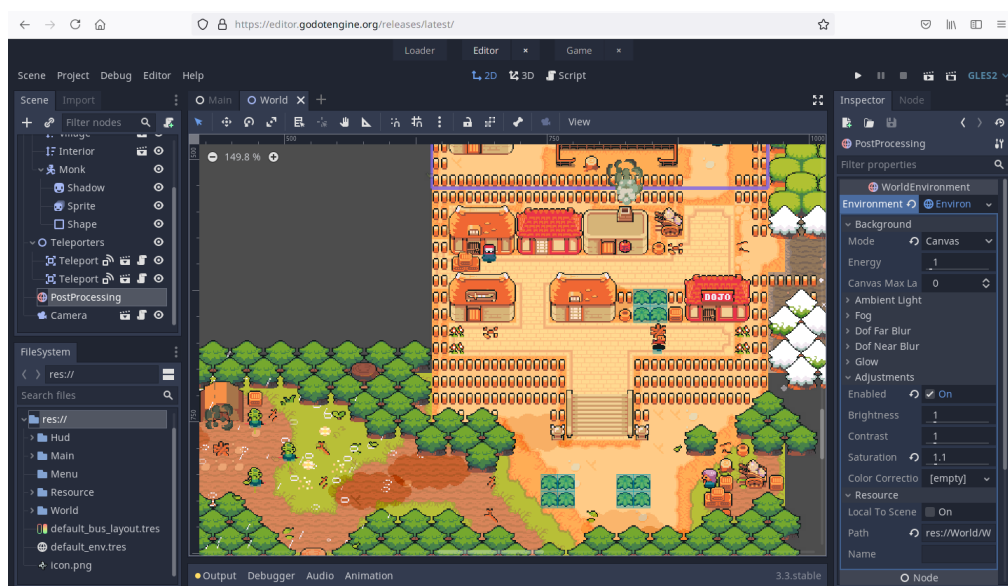


Рис. 12 – Середовище розробки Godot

Робота з графікою в Godot відбувається в окремих підсистемах для 2D і 3D проєктів. Але даний рушія дає змогу зручно комбінувати між собою різновимірні

елементи з допомогою системи ViewPort Node. Також Godot містить вбудовану систему анімації, що дозволяє анімувати будь-який параметер для будь-якого об'єкта на сцені, а також викликати методи з кожного ключа анімації, що значно спрощує анімацію складних сцен.

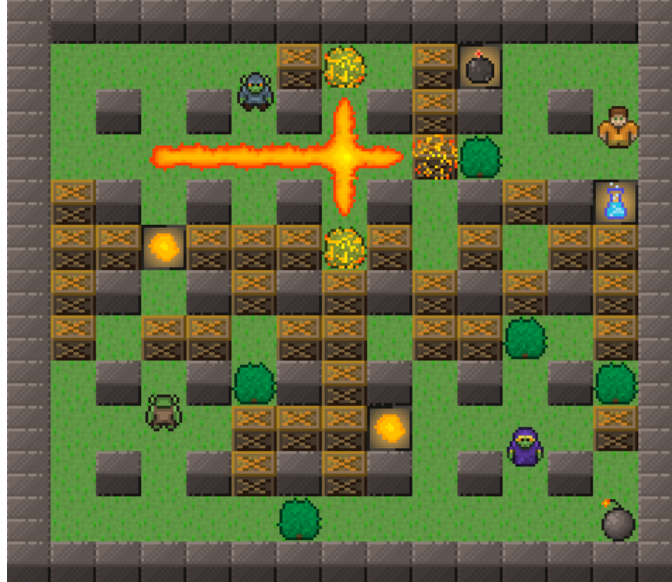


Рис. 13 – Duna Dungeons



Рис. 14 – Tanks of Freedom



Рис. 15 – Minilens

З даних зображень можна побачити, що на сьогоднішній день Godot – не конкурент Unity та Unreal Engine в плані функціоналу, але варто пам'ятати, що він був випущений тільки 10 років тому та все ще проходить фазу розвитку.

РОЗДІЛ 3: МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Математичні операції в проекті

Підчас виконання дипломної роботи виникла проблема числового підрахунку розуміння іноземних слів користувача та оцінювання його знань. У зв'язку з цим була складена формула для обрахунку складності слова. Суть полягає в тому, що слова самі по собі мають різну кількість символів, подвоєння, чергування приголосних, тощо. І від таких нюансів має залежати мінімальна необхідна кількість балів для того, щоб вважати, що дане слово засвоєне. Наприклад, якщо людина ввела слово з семи - десяти символів з другої спроби, то це добре, але якщо стільки ж спроб потрібно на слово з трьох символів, то користувач його точно не знає.

Сама формула обрахунку оцінювання:

$$R = \frac{s + ac + dv * 1.5}{\frac{tc}{rt}} \quad (3.1)$$

R – оцінка відповіді,

s – кількість символів,

ac – кількість чергувань приголосних,

dv – подвоєння голосних,

tc – кількість спроб,

rt – час на переклад у секундах

Результат даної формули буде заокруглюватись до цілого числа та перевірятись, якщо він вийде менше 2 балів, значить користувач слово знає погано чи не знає взагалі.

РОЗДІЛ 4: ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Послідовний опис розробки застосунку.

Для розробки мобільного застосунку потрібно встановити ігровий рушій Unity та середовище для розробки Visual Studio.

В Unity потрібно створити 3D проект. Та підключити бібліотеки, які будемо використовувати в процесі Zenject та UiToolkit.

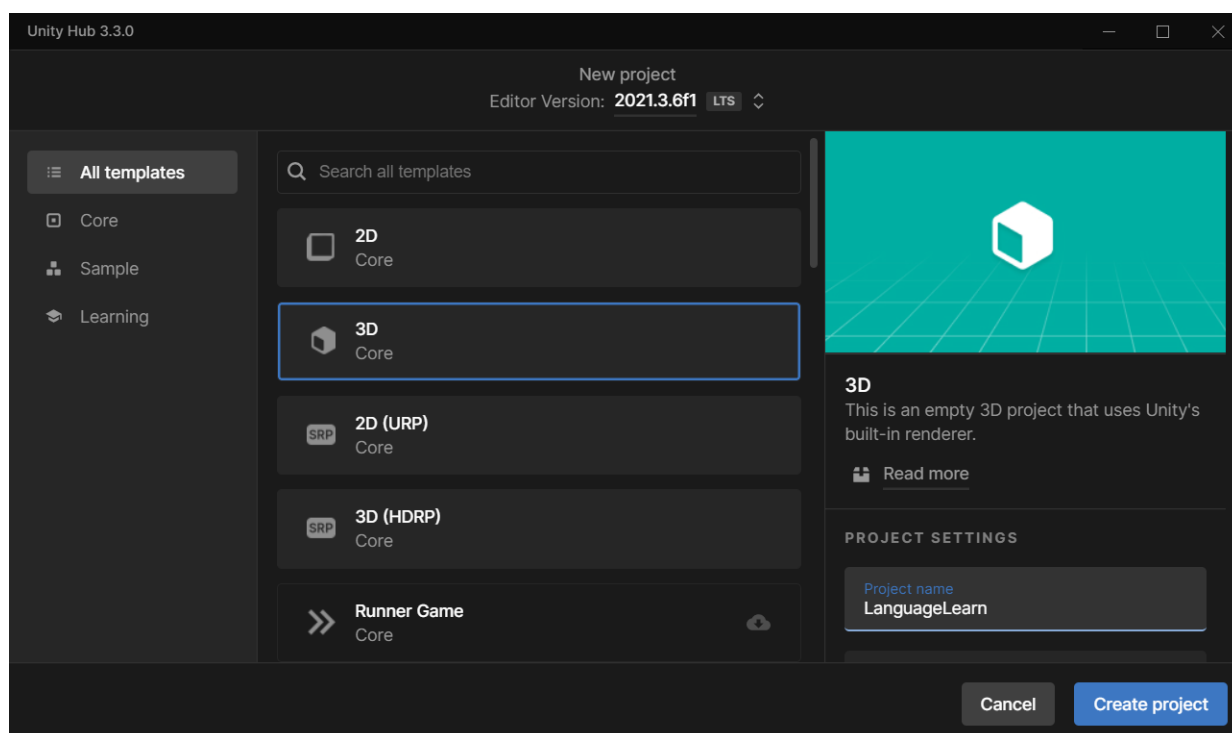


Рис.16 – Створення 3D проекту

UiToolkit – це новий інструмент в Unity, який дозволяє розробляти графічний інтерфейс за схожим методом, як WinForms та Web інтерфейси. Готові інтерфейси зберігаються в ихml файлах та запускаються з допомогою об'єкта з скриптом UiDocument. Даний інструмент дозволяє легко створити гнучкий та динамічний інтерфейс, оскільки ихml файли можна кастомізувати з допомогою uss(Unity Style Sheet) файлів, що працюють за принципом каскадних таблиць стилів.

Zenject, в свою чергу, дозволяє використовувати підхід ін'єкції залежностей, притаманний Asp.Net фреймворку, в роботі з Unity скриптами.

Extenject Dependency Injection IOC



Mathijs Bakker

★★★★★ (39) | ❤️ (708)

FREE

👁️ **330 views** in the past week

Add to My Assets



License agreement [Standard Unity Asset Store EULA](#)

License type [Extension Asset](#)

File size **12.0 MB**

Latest version **9.2.0**

Рис.17 – Додаємо Zenject в AssetStore

Після цього можна буде імпортувати даний asset в Package Manager

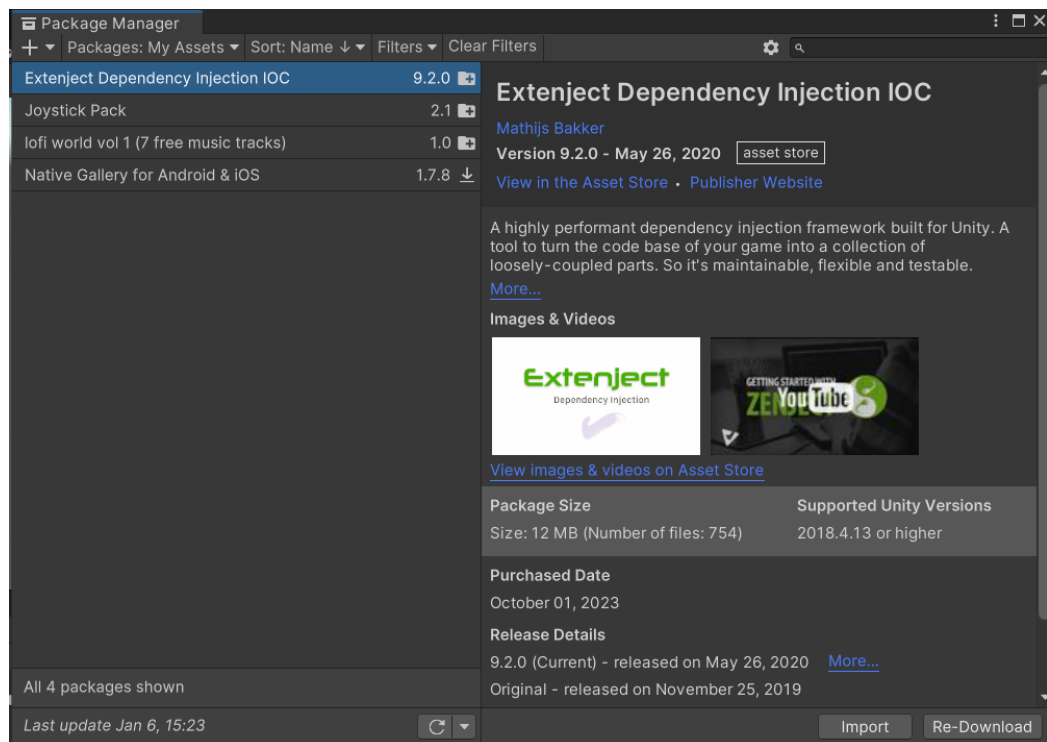


Рис.18 – Імпорт Zenject в проект

Після цього бібліотеку можна використовувати в проекті. Вона дозволить нам імплементувати потрібні залежності в об'єкти класів без утворення сильних зв'язків

між класами, що дозволить в свою чергу упростити модифікацію проекту та додавання нових модулів.

З версії Unity 2022 UiToolkit вбудований в сам ігровий рушій і має окремі інструменти для створення та налагодження користувацького інтерфейсу.

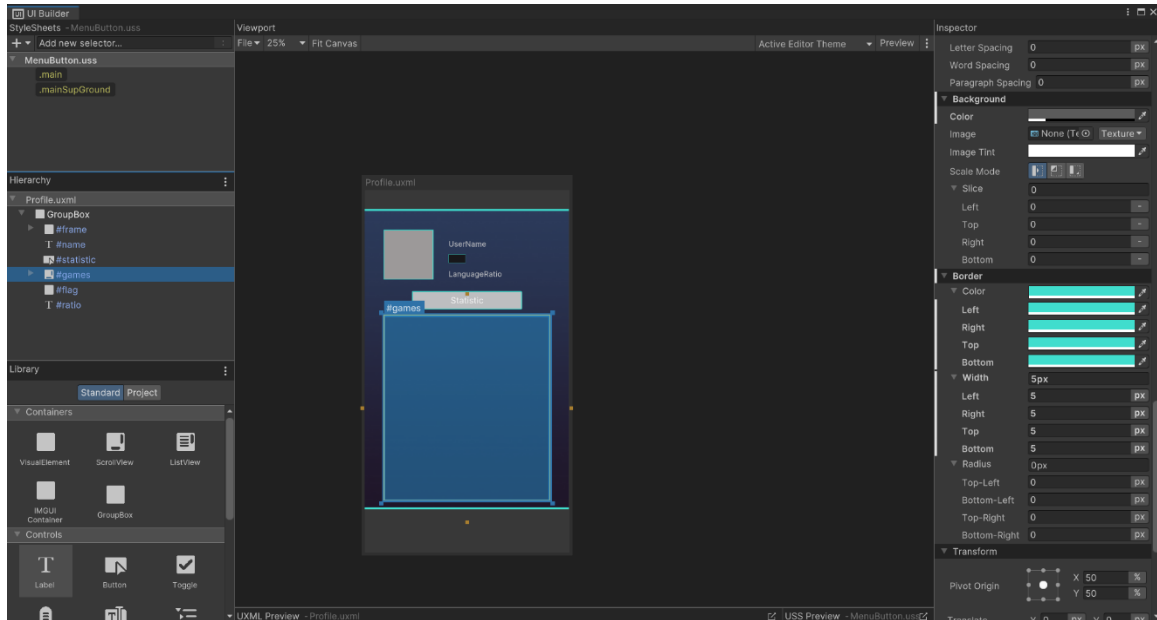


Рис.19 – Робоче середовище UiToolkit

В даному інструменті можна вибрати розмір вікна, а також розмістити там необхідні елементи. Стилі елементів можна задати кожному елементу окремо чи загальною групі за принципом каскадних таблиць стилів.

```
1  .main {
2      border-left-width: 5px;
3      border-right-width: 5px;
4      border-top-width: 5px;
5      border-bottom-width: 5px;
6      border-left-color: rgba(92, 91, 91, 255);
7      border-right-color: rgba(92, 91, 91, 255);
8      border-top-color: rgba(92, 91, 91, 255);
9      border-bottom-color: rgba(92, 91, 91, 255);
10     border-top-left-radius: 30px;
11     border-bottom-left-radius: 30px;
12     border-top-right-radius: 30px;
13     border-bottom-right-radius: 30px;
14     position: absolute;
15     color: rgba(31, 30, 30, 255);
16     -unity-font-style: normal;
17     font-size: 12px;
18 }
```

Рис.20 – Робота з Unity Style Sheet(uss) таблицями

Готовий елемент інтерфейсу можна як і задати вікну напряму, так і використати для подальшої передачі в інший елемент з допомогою коду. Для

кожного з двох випадків буде своє застосування. В першому випадку функціонал можна додати мовою C#, викликавши головний контейнер об'єкта та визначивши івенти для необхідних подій, доступуючись через UiDocument.

```
private void Start()
{
    var root = GetComponent<UIDocument>().rootVisualElement;
    _buttons = root.Query<Button>().ToList();
    for (int i = 0; i < _buttons.Count; i++)
    {
        if (_buttons[i] != null)
        {
            var index = i;
            var button = _buttons[i];
            _buttons[i].RegisterCallback<ClickEvent>(
                ClickEvent e =>
                {
                    ChangePanel(index);
                });
            _buttons[i].RegisterCallback<MouseEvent>((MouseEvent e) =>
            {
                button.style.backgroundColor = _selected;
            });
            _buttons[i].RegisterCallback<MouseEvent>((MouseEvent e) =>
            {
                button.style.backgroundColor = _normal;
            });
        }
    }
}
```

Рис.21 – Реєстрація івентів для переходу панелей меню

В другому випадку шаблон Ui передається, як об'єкт класу VisualTreeAsset і може клонуватись та вставлятись в інші елементи інтерфейсу.

```
1 reference
private void OutputUserData()
{
    if (_userController != null && _userController.User != null)
    {
        var root = GetComponent<UIDocument>().rootVisualElement;
        root.Q<VisualElement>("image").style.backgroundImage = _userController.User.Image;
        root.Q<Label>("name").text = _userController.User.Name;
        var currentLanguage = _userController.User.Languages[_userController.User.CurrentLanguage];
        root.Q<VisualElement>("flag").style.backgroundImage = currentLanguage.Flag;
        root.Q<Label>("ratio").text = currentLanguage.Ratio.ToString();

        GetComponent<UIDocument>().rootVisualElement.Q<ScrollView>("games").Add(_gamesPanel.CloneTree());
    }
}
```

Рис.22 - Вивід даних в UI

В даному проекті у нас буде зберігатись лише зв'язок з серверною частиною та реакція на відповіді сервера, у вигляді маніпуляцій з UI. Основна логіка проекту, а також бази даних зберігатимуться на сервері. Це гарантує захист програми від

впливу шкідливого програмного забезпечення, яке може впливати на ігровий досвід інших користувачів.

Зв'язок з сервером відбувається з допомогою RestAPI та WebSockets. Запити до API використовуються для збору та зміни базової інформації про користувачів, такої як логін, пароль, зображення, тощо. Ці дані зберігаються в SQL таблицях і відповідають за основні параметри користувача. В свою чергу, сокети відповідають за обмін даними під час ігрового процесу. Туди передаються дані, які не будуть зберігатись в БД, а лише в стеку пам'яті, на час гри. Сокети вказують серверу ввід користувача та, відповідно до команд, повертають обробку логіки вводу, яку було винесено на сервер.

Сам сервер розроблений з використанням технологій ASP.Net та EntityFramework. ASP.Net необхідний для побудови контролерів, до яких ми будемо звертатись з допомогою RestAPI запитів.

```
[HttpPost("Registration")]
0 references
public async Task<ActionResult<bool>> Registration([FromBody] UserData userData)
{
    > if (_userService.GetUserByEmail(userData.Email) != null)
    >     return BadRequest();
    > User user = new User()
    >     {
    >         Name = userData.Name,
    >         Email = userData.Email,
    >         Password = userData.Password,
    >         Image = userData.Image
    >     };
    > await _userService.InsertUserAsync(user);
    > return Ok();
}
```

Рис.23 – Контролер реєстрації

Після запити від мобільного застосунку, контролер звертається до методів відповідних сервісів, які вже обробляють винятки та напряму використовують репозиторій – клас, що представляє базу даних і оперує моделями. Моделі – це класи, які відповідають рядкам запису в базі даних.

```

30 references
public class User : Entity
{
    1 reference
    → public string Email { get; set; }
    1 reference
    → public string Password { get; set; }
    3 references
    → public string Name { get; set; }
    2 references
    → public string Image { get; set; }
}

```

Рис.24 – Модель користувача

Сам мультиплеєр розділяється на систему для пошуку суперників та розподіл кімнат. Коли підключається достатня кількість користувачів для гри, створюється ігрова кімната з користувачами та типом гри, який вони шукали. В залежності від типу гри будуть прийматись різні json файли з мобільного застосунку та відправлятись різні відповіді у вигляді тих же json-файлів.

```

1 reference
public static void Listen(this TcpClient client, Action<string> messageReceiver = null, Action onDisconnect = null)
{
    try
    {
        while (true)
        {
            var stream = client.GetStream();
            byte[] buffer = new byte[1024];
            int bytes = stream.Read(buffer, 0, buffer.Length);
            string message = Encoding.UTF8.GetString(buffer, 0, bytes);
            if (message == string.Empty)
            {
                throw new Exception("Broken connection");
                Console.WriteLine(message);
                messageReceiver?.Invoke(message);
            }
        }
    }
    catch (Exception e)
    {
        client.Close();
        Console.WriteLine(e.Message);
        onDisconnect?.Invoke();
    }
}

```

Рис.25 – Розширення клієнта для прослуховування

4.2 Тестування програми.

При запуску програми ми відправляємо запит на сервер з GoogleId користувача. Після цього на стороні сервера отримуємо Email за GoogleId і повертаємо користувача, його ім'я, заставку та рейтинг. Після цього дані

користувача зберігаються в глобальну змінну, яка буде зберігатись в проекті для подальшого використання.

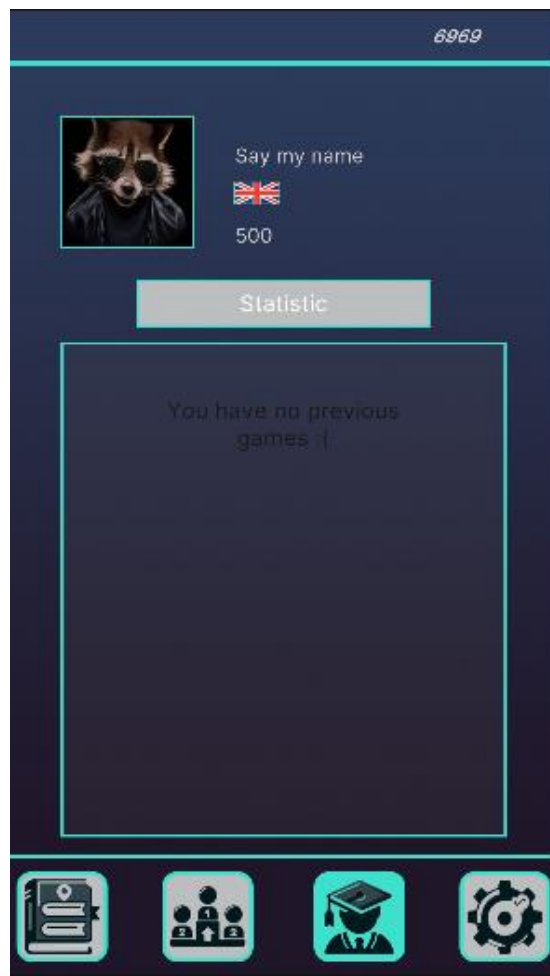


Рис.26 Профіль користувача

До даного проекту можна додавати будь-які ігри, які будуть працювати на віддаленому сервері. Для тестової перевірки роботи сервера була створена вікторина, яка перевіряє обмін даних з сервером.

По-перше сервер відправляє набір символів, з яких буде формуватись колесо для створення слів. Ідея для даної механіки була запозичена з кросворду Words of Wonders.



Рис.27 Ігрова механіка “колесо комбінацій”

Всі символи завантажені з ігрової кімнати сервера, на якій вже записані слова, які можна скласти з даних символів.

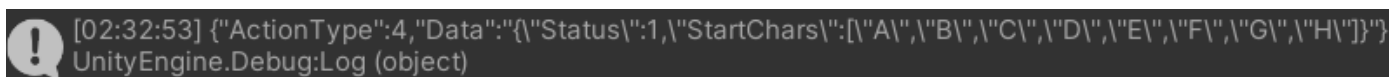


Рис.28 JSON-файл з символами колеса

Після цього користувач може вводити комбінації, які відправляються на сервер для подальшої перевірки. Дані комбінації також передаються в JSON, який перетворюється на сервері та шукає відповідники до комбінації в словнику.

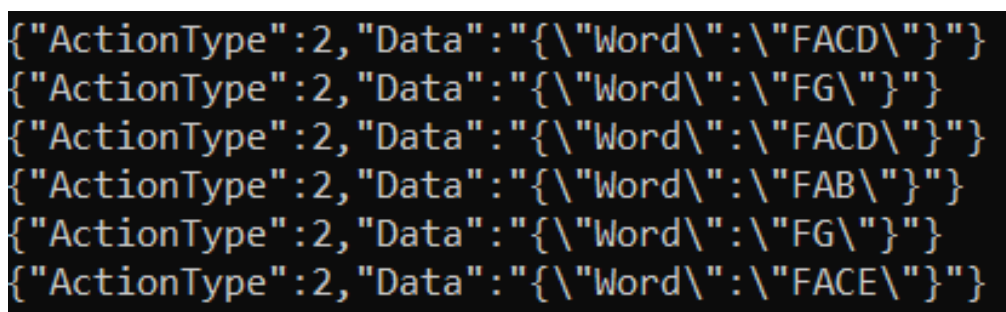


Рис.29 – Дані кожної комбінації

Як можна зрозуміти з попереднього рисунка, пошук за пошук слів відповідає тип дії під номером 2. Номери 0 та 1 зарезервовані для з'єднання з сервером та відновлення з'єднання, 3 – перевірка перекладу слова, а 4 – заповнення колеса. Кожному типу дії відповідає клас-модель, який містить відповідну інформацію, що буде десеріалізована та записана в рядок до подальшої передачі.

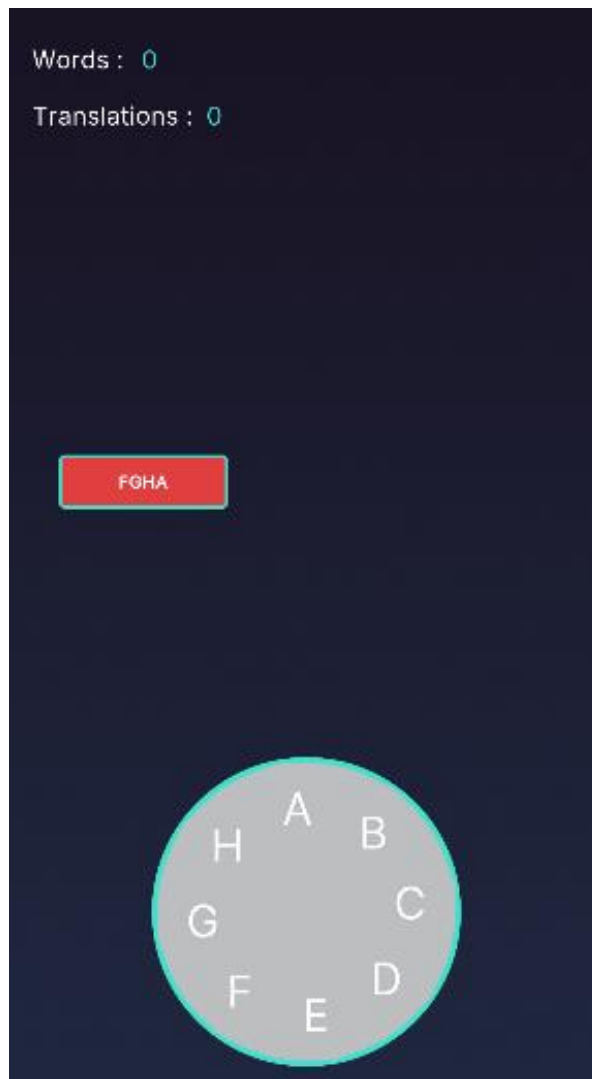


Рис.30 – Реакція UI на неправильне слово

Якщо даного слова немає в списку, то повертається рядок з Status дорівнює 0.

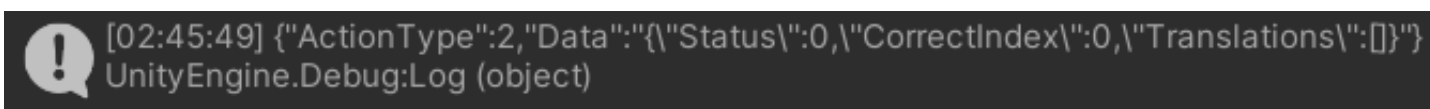


Рис.31 – Відповідь на неправильне слово

При коректному вводі повертається індекс правильного перекладу та кілька варіантів перекладу на вибір. Після вибору перекладу дані відправляються на сервер і записується результат вводу, який збережеться в JSON-файл після гри, визначивши, які слова потрібно буде повторити чи вивчити.

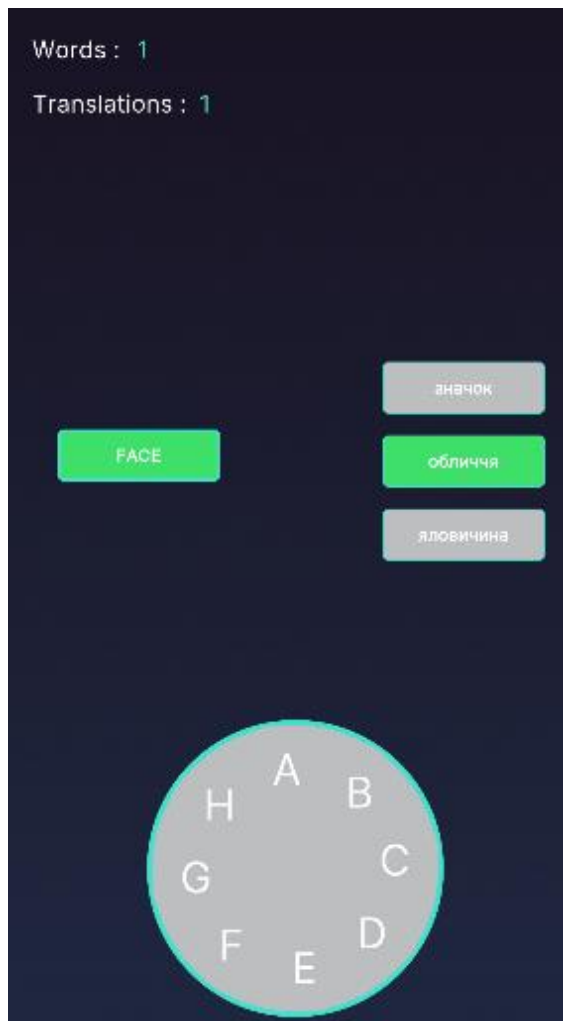


Рис.32 Правильний ввід слова та переклад

РОЗДІЛ 5: РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

5.1. Опис ідеї проекту

Головною ідеєю проекту є розробка платформи для вивчення іноземних мов, яка буде охоплювати максимально широку цільову аудиторію та містити навчальні комплекси, підібрані для різних вікових категорій та верств населення. Суть даної ідеї полягає в розробці застосунка таким чином, щоб користувачам було легко там орієнтуватись, цікаво знаходитись та продуктивно навчатись, не залежно від потреб людини – через розваги чи через інтенсивне навчання. Для таких цілей програма має містити взаємодії між гравцями на рівні навчання, поглиблені курси та можливість самостійно аналізувати свої успіхи у вивченні мов. В пріоритеті цей застосунок можна розширити до загальної навчальної платформи, що дозволить створювати курси не лише з іноземних мов, а й з програмування, математики, історії, тощо.

Напрямки застосування такі ж, як в інших освітніх застосунках - сфера освіти. Застосунок може бути, як самостійна навчальна програма, так і платформа для розміщення матеріалів викладачами чи школами. Поширення проекту може стосуватись не тільки смартфонів з операційною системою Android, а й техніки на базі IOS, а також веб-сайтів та застосунків. Це має змінити ставлення до освітніх мобільних програм з допоміжних інструментів на потужні засоби для освіти та викладацької діяльності.

Користувачі проекту зможуть самостійно вибирати собі тип і темп навчання, аналізувати свої досягнення з допомогою статистики та рейтингової системи, закріплювати матеріал граючи з друзями чи штучним інтелектом, а також засвоювати теми легко та невимушено. Застосунок зробить навчання доступним для широкої аудиторії будь-де та будь-коли, а також об'єднає переваги та усуне недоліки своїх конкурентів. Для викладачів даний проект дозволить формувати власні курси та полегшувати викладацьку діяльність, а також надасть повний аналіз активностей та успіхів студентів.

Основною відмінністю від конкурентів на ринку буде відсутність строгих покупок, які впливатимуть на тривалість чи ефективність навчання. Система

покупок впливатиме лише на якість аналізу, відсутність реклами та косметичні речі. Також інноваційністю можна вважати введення справжніх мультиплеєрних механік в освітню платформу та рейтингової системи оцінювання. Такий підхід дозволить посилити здорову конкуренцію між користувачами та зробити навчальний процес різноманітнішим.

5.2. Аналіз технологічних можливостей реалізації проекту

Оскільки, на даний момент, розглядається випуск проекту лише на сенсорні телефони, то найкращим варіантом буде використати ігровий рушій Unity. Причина полягає в тому, що для розробки мобільних ігор, на даний момент, не існує рушія, який здатний замінити Unity на даному ринку. Також одним з важливих факторів вибору саме Unity полягає в тому, що він дозволяє розробляти кросплатформенні проекти, які легко пристосувати до будь-якої операційної системи. Для розробки сервера вибраний фреймворк Asp.Net та платформа .Net Core мови C#. Причиною такого вибору стала зручність розробки, оскільки розробка на Unity також відбувається на мові C# та з допомогою платформи .Net Core. Така комбінація дозволить впроваджувати зміни в проект невеликій кількості розробників, оскільки один розробник зможе працювати, як і з сервером, так і з мобільним застосунком. Щодо Asp.Net, то даний фреймворк дозволяє легко зробити API для зв'язку з застосунком, а також в купі з Entity Framework він забезпечить надійний доступ до реляційних баз даних і безпечний обмін інформацією.

Дуже важливою складовою даного стеку технологій є те що вони безкоштовні та максимально доступні для розробника. Звичайно, ігровий рушій Unity має платну підписку та забирає відсоток з заробітку розробника, але ці всі мінуси неактивні до тих пір, поки застосунок не почне приносити прибуток, а отже, почати розробку можна з мінімальними вкладеннями капіталу.

5.3. Аналіз ринкових можливостей запуску стартап-проекту

Як уже згадувалось в розділі 1.1 “Огляд проблемної області”, за останні 10 років, попит на вивчення іноземних мов зріс в середньому в 4 рази, а кількість користувачів мобільними застосунками для вивчення іноземних мов нараховується сотнями мільйонів. З огляду на це, можна припустити, що дана ніша має великий

попит та є прибутковою. Проте є один нюанс, який полягає в тому, що в сфері таких застосунків вже є свої гіганти, які мають величезну популярність і привертають до себе всю увагу. Це впливає на новостворені застосунки в тому плані, що їх можуть не помітити на старті користувачі, тому вони просто не зможуть вирости. Навіть попри те, що проєкт переважає конкурентів функціоналом і має унікальні характеристики, цього може бути недостатньо для захоплення своєї частини ринку. Як наслідок, на старті запуску доведеться влити великі кошти на рекламу продукту та його просування в рейтингах маркетів.

Хоча конкуренція на ринку освітніх застосунків дуже потужна, вона не є непереможною та вирішується хорошою маркетинговою стратегією. Основна проблема виникне вже на етапі росту, оскільки застосунок може привернути увагу конкурентів, які не захочуть ділитись своїм місцем на ринку і можуть перешкоджати маркетинговій стратегії шляхом викуплення рекламних місць, а також перетягуванням уваги на себе. На випадок подібної ситуації варто розробити стратегію захисту ще на етапі планування, яку можна буде втілити зразу ж, як тільки продукт почне приносити перші результати.

5.4. Розроблення ринкової стратегії проєкту

На початок розвитку проєкту варто скористатись стратегією концентрованого маркетингу, щоб швидко набрати популярність та охопити максимально велику частку ринку. Я вважаю, що початковою аудиторією мають стати підлітки та доросла аудиторія до 30 років, що обрала вивчення іноземних мов, як хоббі у вільний від навчання та роботи час, оскільки навчання побудоване на конкуренції з найбільшою ймовірністю зацікавить саме їх. Після цього варто перейти на диференційовану стратегію маркетингу, починаючи поширювати свій вплив на попередню аудиторію та залучаючи дитячу, оскільки принцип навчайся-граючи найбільше підходить для неї. Такий початок забезпечить міцну позицію проєкту на шляху до розвитку в навчальну платформу та залучення приватних репетиторів та шкіл до роботи на платформі.

5.5. Маркетингова програма стартап-проєкту

Ще в процесі розробки проєкту можна почати набір кістяка аудиторії, ведучи соціальні мережі, де будуть набиратись бета-тестувальники проєкту, відбуватись комунікація з підписниками. Це допоможе кількома способами: перший – комунікація дасть можливість перевірити ринок та дізнатись з перших уст, чого не вистачає користувачам подібних застосунків, другий – збереться колектив, який в майбутньому зможе надати відгуки про бета-версію проєкту та остання перевага – в день запуску проєкту підписники стануть органічними користувачами готового проєкту, що швидко підніме програму в рейтингах маркетів.

Варто відзначити, що продукти такого типу на ринку знаходяться в доступі free-to-play. Тобто монетизація з них надходить не напряду від покупки одиниці продукту, а від внутрішньо-ігрових надходжень, а також з реклами. В зв'язку з цим проєкт має бути також free-to-play, оскільки ціна на скачування невідомого продукту відштовхне майбутніх користувачів.

В подальшому для захоплення стартової цільової аудиторії знадобиться закупівля реклами в тематичних блогерів, соціальних мереж, тощо. Це дозволить збільшити кількість активних користувачів та розвинути застосунок до другої стадії.

Після цього можна випустити окремий застосунок для дітей, який буде рекламуватись за рахунок основного застосунку та докупки орієнтованої на дітей реклами. Нову програму треба буде зробити привабливою для дітей: яскраішою, інтерактивнішою, з меншою кількістю конкуренції. Хорошим варіантом на цій стадії буде створити та популяризувати маскота застосунку.

І на кінцевому етапі вже можна запускати оголошення про створення навчальної платформи. Успіх в двох перших етапах значно збільшить можливість того, що навчальні заклади почнуть звертатись самі, щоб навчати учнів на новій платформі.

5.6. Висновки до розділу

Попри велику конкуренцію на ринку, створений продукт має хорошу можливість закріпитись там та отримати прибуток. Сам по собі заробіток з проєкту буде надходити з кількох джерел, типових для ринку мобільних застосунків. До них відноситься перегляд реклами, внутрішньо-ігрові покупки, такі як відключення

реклами, преміум статус з доступом до розширеної статистики, додаткових ігор та матеріалів і покращеної системи навчання та косметичних наборів.

Оскільки, продукт потребуватиме капіталовкладень на початковому етапі, до того, як він почне приносити реальні прибутки, можливо, доведеться шукати інвестиції для його розвитку. Це може спричинити додаткове навантаження на компанію та команду через те що доведеться стабільно показувати хороші результати, бо є ймовірність залишитись без фінансування та збанкрутувати так і не досягнувши успіху. А таке навантаження потребуватиме розширення штату працівників, таких як програмісти, дизайнери та маркетологи. Також необхідним буде гейм-дизайнер, який зможе генерувати нові ідеї та відкидати зайві, що збереже час та підвищить ймовірність успішного закріплення продукту на ринку.

ВИСНОВКИ

Створено мобільний застосунок для вивчення іноземних мов. Додано можливість брати дані користувача з сервера та передавати їх на мобільний застосунок з допомогою HTTP запитів та API. Реалізовано мультиплеєр типу Клієнт-Сервер з допомогою бібліотеки WebSockets. Налагоджена передача даних між мультиплеєром та користувачем з допомогою JSON-файлів. Створено UI мобільного застосунку з допомогою бібліотеки UnityEngine та інструментів UiToolkit.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Джеремі Гібсон Бонд. Unity і C#. Геймдев від ідеї до реалізації, друге видання, 2022
2. Jeffrey Richter. CLR via C#, 4 edition - November 15, 2012
3. Joe Hocking. Unity in action: Multiplatform game development in C#, Third edition - February 8, 2022
4. Matt Smith. Unity 2021 Cookbook: Over 140 recipes to take your Unity game development skills to the next level, 4th Edition, September 6, 2021
5. Thomas Finnegan. Learning Unity Android Game Development, April 28, 2015
6. Патерни проектування - <https://refactoring.guru/uk/design-patterns>
7. Документація Asp.Net - <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>
8. Документація Unity - <https://docs.unity3d.com/Manual/index.html>
9. Документація .Net - <https://docs.microsoft.com/en-us/dotnet/fundamentals/>
10. Entity Framework документація - <https://learn.microsoft.com/en-us/ef/>

ДОДАТКИ

```
internal static class TcpClientExtention
{
    public static void SendMessage(this TcpClient client, string message)
    {
        var stream = client?.GetStream();
        if (client.Connected && stream != null)
        {
            byte[] response = Encoding.UTF8.GetBytes(message);
            stream.Write(response, 0, response.Length);
        }
    }

    public static void Listen(this TcpClient client, Action<string> messageReceiver = null, Action onDisconnect = null)
    {
        try
        {
            while (true)
            {
                var stream = client.GetStream();
                byte[] buffer = new byte[1024];
                int bytes = stream.Read(buffer, 0, buffer.Length);
                string message = Encoding.UTF8.GetString(buffer, 0, bytes);
                if (message == string.Empty)
                    throw new Exception("Broken connection");
                Console.WriteLine(message);
                messageReceiver?.Invoke(message);
            }
        }
        catch (Exception e)
        {
            client.Close();
            Console.WriteLine(e.Message);
            onDisconnect?.Invoke();
        }
    }
}

internal class Room
{
    public Guid Id { get; private set; }
    private Game _game;
    private List<ConnectionData> _players;
    private bool _running;
    public Room(List<ConnectionData> players)
    {
        _players = players;
        Id = Guid.NewGuid();
        _players.ForEach(p =>
        {
            var room = new RoomSerializationData()
            {
                Id = Id,
                PlayerId = p.Id,
                Players = _players.FindAll(ps => ps.Id != p.Id).ToList()
            };
            p.Client.SendMessage(JsonConvert.SerializeObject(new ActionInfo() { ActionType = ActionType.Connect,
                Data = JsonConvert.SerializeObject(room) }));
        });
    }
}
```

```

        ListenPlayer(p);
    });

    _game = new TranslationBattles(_players);
    _game.Start();
}

public bool Reconnect(ConnectionData player)
{
    var searchedPlayer = _players.FirstOrDefault(p => p.Id == player.Id);
    searchedPlayer.Client = player.Client;
    searchedPlayer.Client.SendMessage("Reconnected");
    _running = true;
    return searchedPlayer != null;
}

private void ListenPlayer(ConnectionData player)
{
    new Task(() =>player.Client.Listen(
        (message) =>
        {
            _game.DoAction(player, message);
        },
        () =>
        {
            _running = false;
        }
    )).Start();
}
}

```